UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
08/880,616	06/23/97	COHEN	M Y0997-111

LM51/1001

DOUGLAS W CAMERON
IBM CORPORATION
P O BOX 218
YORKTOWN HEIGHTS NY 10598

EXAMINER
FOURSON, G

ART UNIT	PAPER NUMBER
2755	

DATE MAILED:

10/01/99

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

1- File Copy

Office Action Summary

	Application No. 08/880,616	**Applicant(s)** Cohen et al.
	Examiner Gary Fourson	**Group Art Unit** 2755

☒ Responsive to communication(s) filed on *Sep 1, 1999* .

☐ This action is **FINAL**.

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire ___*3*___ month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Disposition of Claims

☒ Claim(s) *1-18* is/are pending in the application.

   Of the above, claim(s) _____ is/are withdrawn from consideration.

☐ Claim(s) _____ is/are allowed.

☒ Claim(s) *1-18* is/are rejected.

☐ Claim(s) _____ is/are objected to.

☐ Claims _____ are subject to restriction or election requirement.

Application Papers

☐ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The drawing(s) filed on _____ is/are objected to by the Examiner.

☐ The proposed drawing correction, filed on _____ is ☐approved ☐disapproved.

☐ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

   ☐ All ☐ Some* ☐ None of the CERTIFIED copies of the priority documents have been

      ☐ received.

      ☐ received in Application No. (Series Code/Serial Number) _____ .

      ☐ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   *Certified copies not received: _____ .

☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

☒ Notice of References Cited, PTO-892

☐ Information Disclosure Statement(s), PTO-1449, Paper No(s). _____

☐ Interview Summary, PTO-413

☐ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Notice of Informal Patent Application, PTO-152

--- *SEE OFFICE ACTION ON THE FOLLOWING PAGES* ---

DETAILED ACTION

Claim Rejections - 35 USC § 103

1.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth
> in section 102 of this title, if the differences between the subject matter sought to be patented and the prior
> art are such that the subject matter as a whole would have been obvious at the time the invention was
> made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall
> not be negatived by the manner in which the invention was made.

2.   **Claims 1, 2, 4-16, and 18 are rejected under 35 U.S.C. 103(a) as being**

unpatentable over Cameron et al. (5,325,526) in view of Admitted Prior Art

(Applicant's Specification page 2 line 20 through page 3 line 7).

        With respect to claim 1, Cameron et al. teaches computing nodes (processors P1,

P2, and P3) with a plurality of local processes (column 2 line 30, "a plurality of computer

application programs ...", lines 39-40, "task scheduling system in a multicomputer having

nodes ..."; column 7 line 36, "... more than one application program may be loaded on a

single node.", lines 53-57, "A partition in the present invention is an object comprising a

plurality of items of information and optionally related processing functions for

maintaining a logical environment for the execution of tasks of one or more application

programs."), scheduler means (scheduler 410, 510, or 612), and dynamically creating a

schedule of a plurality of tasks utilizing priorities (The column 14 lines 33-42: interactive

scheduling The flexible, hierarchical division of the processing power into partitions and

layers, to which applications are assigned. In column 15 line 3 Cameron et al. state,

"Layers serve as the basis for scheduling ..." Column 15 continues that layers are

scheduled in time quanti where one or more applications [and/or sub-partitions] assigned

to a particular layer are made active for a time quantum. The layer is divided into sets of

processing nodes where each set is assigned to a particular application. When a layer is

active for a particular time quantum, each node in the set of nodes of the application to

which they belong schedules the task of that application in a routine, prior art

multitasking method. Partitions acquire the priority of the highest priority application

contained, and the current priority may dynamically change, see column 9 lines 56 et.

seq. It is the Examiner's duty and obligation to Applicant and to the public to give

Applicant's claims their broadest reasonable interpretation. Therefore, it would have been

obvious to one of ordinary skill in the art at the time the invention was made to utilize the

priority and maximum priority fields 918 and 920 respectively to dynamically schedule

tasks from multiple applications for round robin execution in the time quanti, because the

linked lists of partitions, applications, and layers shown in Figures 9-12 elaborated in

column 10 utilize the priority/maximum priority fields.), and a local scheduler (Figure 4

shows and column 5 line 58 through column 7 line 29 elaborates a prior art scheduling

system. Column 6 lines 16-21 state, "If another task in a scheduling queue is ready to

run, the currently executing task is suspended and the next task in the scheduling queue of the highest priority is activated."), and means for ascertaining which process(es) are assigned to the tasks (Column 9 lines 28-31, "The layer data structure 738 comprises information including identity of the nodes of the partition that are allocated by a list of consumers to which the layer points." The layer data information is used to correspond applications assigned to a set of nodes. The processes of each application in the layer are activated for the execution time quantum, thus in any one time quantum the correlation of tasks to processes is merely based on the distribution of the nodal processing power in that active layer.). However, Cameron et al. does not appear to teach means for prioritizing the processes according to the prioritized schedule.

Applicant has divulged on pages 2-3 that the AIX™ operating system assigns a common priority to the process(es) required for (or correlated to) a task. Having the processes, associated with individual tasks, assigned priorities corresponding to the priorities of the schedule would have been a highly desirable feature in the art. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the local scheduling/correlating means of IBM into the task scheduling system of Cameron et al., because prioritizing local processes according to the task correlation would have been expected to result in higher cache preloading efficiency.

With respect to claim 11, Cameron et al. in view of Admitted Prior Art as

modified for the rejection of claim 1 teaches the limitations of claim 11 substantially as

claimed except providing application information to the scheduler means: Cameron et al.

teaches in columns 7 and 8 an allocator and scheduler component (612 and 710). Figure

7 shows procedures (720, 722, 724, 726, 728, 740, 742, and 744) for the Allocator and

Scheduler. Block 736 represents application data which inherently must have been

transmitted to the allocator/scheduler.

As to **claim 2,** Cameron et al. in view of Admitted Prior Art as modified for the

rejection of claim 1 teaches the limitations of claim 11 substantially as claimed except

said computing node comprising an operating system for "receiving input" from the

prioritizing means and "directing said assigned processes" to execute tasks in a prioritized

order: Cameron et al. in Figures 4 and 5 show a prior art task scheduler. Column 5 last

paragraph elaborates stating that each scheduler comprises operating system software

responsible for controlling the execution of a plurality of tasks. It would have been

obvious to one ordinarily skilled in the art at the time the invention was made for the OS

to receive information about the execution of the plurality of tasks, because Cameron et

al. states in column 6 lines 28-31, "Interactive scheduling using Unix, or other operating

systems in a single processor environment, is well known to those of ordinary skill in the

art."

As to **claim 4**, Cameron et al. in view of Admitted Prior Art as modified for the

rejection of claims 1 and 2 teaches the limitations substantially as claimed except

application coordinator means for communicating information to said scheduler:

Scheduling information must inherently be obtained by some means in order to produce a

prioritized list of tasks, however in column 8, Cameron et al. teaches on line 2, "The

allocator and scheduler 710 comprises processing logic and data for allocating nodes to

specific application programs and for scheduling applications programs for execution."

The "Make Partition" procedure (720) is the request for the allocator/scheduler to

initialize tasks which as stated on line 18, "are retrieved and loaded into the nodes

associated with the specified partition."

As to **claim 5**, Cameron et al. in view of Admitted Prior Art as modified for the

rejection of claims 1 and 2 teaches the limitations substantially as claimed except said

local processes being adapted to perform tasks in parallel: Also, in column 2 on line 50

Cameron teaches that application programs are allowed to execute on one or more nodes

of a partition. Furthermore, column 7 line 40 states, "...an entire application program is

active at once across all of the nodes on which the application program is loaded." The

multi-node or multi-processor collaborative effort to the processing of a set of tasks or

application program processes is the truest definition of parallel processing. Cameron et

al. in column 1 on lines 26 to 30 indicates that multi-tasking, round robin processing, time

slicing, or parallel processing was well known to one of ordinary skill in the art at the

time the invention was made.

As to claim 6, Cameron et al. in view of Admitted Prior Art as modified for the

rejection of claim 1 teaches the limitations substantially as claimed except said scheduler

means comprising global scheduler means which in turn comprises means for

dynamically scheduling then communicating the schedule to the local scheduler:

Cameron et al. teaches the local nodes are assigned to application programs. The

allocator and scheduler 612 act functionally as a "global scheduler" by controlling and

assigning the appropriate nodes from a particular layer. Column 7 line 50 states, "As will

be described below, allocator and scheduler 612 may and typically does operate with a

plurality of partitions 614." In column 9 on line 50, "In the preferred embodiment,

partition data blocks and application data blocks can be maintained in the same doubly-

linked list." Further down on line 64 it is stated that, "The current priority field 918 may

dynamically change as the priorities of associated application programs or sub-partitions

change priority."

As to claim 7, Cameron et al. in view of Admitted Prior Art as modified for the

rejection of claims 1 and 6 teaches the limitations substantially as claimed except said

local scheduler being adapted to communicate process information to the global

scheduler: Cameron et al. teaches in column 14 lines 12-31 three access modes to the

partition data. They are read, write, and execute access modes allowing or disallowing the ability to run application programs from a partition and to create or remove sub-partitions from a partition. This information is also available to the allocator/scheduler 710. Also, figure 7 shows application data 736 specifically available to the allocator/scheduler.

As to **claim 8**, Cameron et al. in view of Admitted Prior Art as modified for the rejection of claims 1 and 6 teaches the limitations substantially as claimed except the global scheduler also comprising timer means to effect schedule communication: Cameron et al. teaches in column 11 lines 6-11 a time executed field 1021.

As to **claim 9**, Cameron et al. in view of Admitted Prior Art as modified for the rejection of claims 1 and 6 teaches the limitations substantially as claimed except said global scheduler including a local scheduler address table: Cameron et al. teaches in column 13 lines 15-33, "Two hash tables providing a quick look-up mechanism for locating partitions ..."

As to **claim 10**, Cameron et al. in view of Admitted Prior Art as modified for the rejection of claims 1, 2, and 6 teaches the limitations substantially as claimed.

As to **claim 12**, Cameron et al. in view of Admitted Prior Art as modified for the rejection of claims 1 and 11 teaches the limitations substantially as claimed except invoking operating system priorities to schedule tasks in accordance with said prioritized

schedule: The operating system would inherently follow any prioritizing scheme

employed by the programmer or else there would not be any need to incorporate the

local/global scheduling means in the first place.

As to **claim 13**, Cameron et al. in view of Admitted Prior Art as modified for the

rejection of claims 1 and 11 teaches the limitations substantially as claimed except

scheduler means is remote to the node and communicating the schedule to the node:

Cameron et al. shows in Figures 4 and 5 that in prior art methods of task management

systems, the Scheduler 410, 510 can be remotely located from the processors. In column

6 lines 32-45 refer specifically to Figure 5 noting that the scheduler arranges an orderly

schedule for multiple tasks executing on multiple processors. Line 37 mentions a

common memory where the schedule information would be communicated to the three

processing nodes.

As to **claim 14**, Cameron et al. in view of Admitted Prior Art as modified for the

rejection of claims 1, 5, 11, and 12 teaches the limitations substantially as claimed.

As to **claim 15**, Cameron et al. in view of Admitted Prior Art as modified for the

rejection of claims 1, 11, 12, and 14 teaches the limitations substantially as claimed

except communicating task execution information to the scheduler: Cameron et al.

teaches in column 11 lines 6-11 a time executed field 1021. This "execution

information" is part of a process group field 1020 which is updated and available to the scheduler.

As to **claim 16**, Cameron et al. in view of Admitted Prior Art as modified for the rejection of claims 1, 5, 11, 12, 14, and 15 teaches the limitations substantially as claimed except repeating said steps until all tasks have been completed: Cameron teaches recursive scheduling in column 15 on lines 12-14.

As to **claim 18**, Cameron et al. in view of Admitted Prior Art as modified for the rejection of claims 16, 11, and 13 teaches the limitations substantially as claimed except said remotely located scheduler dynamically maintaining a computing node's list: Figure 7, Layer Data -738-; Column 9 lines 28-31, "The layer data structure 738 comprises information including identity of the nodes of the partition that are allocated by a list of consumers to which the layer points."

3. **Claims 3 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cameron et al. as modified by Applicant's Admitted Prior Art as applied to claims 2 and 14 above, and further in view of Ripps (The Multitasking Mindset Meets the Operating System).**

As to **claim 3**, Cameron et al. in view of Admitted Prior Art as modified for the rejection of claims 1 and 2 teaches the limitations of claim 11 substantially as claimed except the operating system being further adapted to interleave local operations with said

tasks: A node or CPU controlled by an operating system would inherently process local operations (e.g. an exception) pertaining to the operating system commands. Ripps teaches on page 9 that C and proprietary OS functions are intermixed in a typical task. Context switches controlled by the operating system are also well known local tasks which are interleaved between the application task execution.

As to **claim 17**, Cameron et al. in view of Admitted Prior Art as modified for the rejection of claims 1, 3, 5, 11, 12, and 14 teaches the limitations substantially as claimed.

4.   Claims 1 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zolnowsky (5,826,081) in view of Custer (Inside Windows NT) and further in view of Admitted Prior Art (Applicant's Specification page 2 line 20 through page 3 line 7).

With respect to claims 1 and 11, Zolnowsky teaches a plurality of tasks (threads) of more than one application (Single processor, multitasking operating systems such as Windows NT described by Custer, see page 83 line 3, routinely schedule threads from more than one application. Adding a multiplicity of processors for greater computing capabilities for multiple applications would have been a highly desirable feature in the multitasking art. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to execute a plurality of tasks from more than one application on the multiprocessor system, because the real time applications of

Zolnowsky are typically partitionable into schedulable entities or mixed-mode

applications, see column 2 lines 35-40.), computing nodes (col. 3 line 52, "multiple

processors P1 through Pn"), a plurality of local processes (Zolnowsky defines tasks as the

unit of resource management and the thread as a single flow-of-control in column 1 lines

31-34. In column 6 lines 63-65 indicate that some threads are only executable on a

particular processor of the multiprocessor assembly. Therefore, there is a process [or task

as Zolnowsky has so defined] that is different on that particular processor from other

processes on other processors.), providing application information to scheduler means

(column 5 lines 26-38), and dynamically creating a schedule of a plurality of tasks

utilizing priorities (col. 6 line 45, "Also, when a thread is made runnable, it is placed on a

dispatch queue, ..."; col 8 line 66, "Real time threads are scheduled strictly on the basis of

their priority ..."), a local scheduler (column 7 line 15, "Each processor ..."), and means

for ascertaining which process(es) are assigned to the tasks (Custer teaches on page 84

that a process must be combined with a thread of execution before it can do any work.

The thread of execution "is the entity within a process that the NT kernel schedules for

execution. Without it, the process's program cannot run." Therefore, it would have been

obvious to one of ordinary skill in the art at the time the invention was made to

incorporate means for correlating or *ascertaining which process(es) are assigned to the*

tasks, because Custer recognized without the correlation between processes and threads

the process or program would not be able to execute.). However, Zolnowsky as modified

by Custer does not appear to teach means for prioritizing the processes according to the

prioritized schedule.

Applicant has divulged on pages 2-3 that the AIX™ operating system assigns a

common priority to the process(es) required for (or correlated to) a task. Having the

processes, associated with individual tasks, assigned priorities corresponding to the

schedule would have been a highly desirable feature in the art. Therefore, it would have

been obvious to one of ordinary skill in the art at the time the invention was made to

incorporate the local scheduling/correlating means of IBM into the task scheduling

system of Zolnowsky, because prioritizing local processes with the current and next task

would have been expected to result in higher cache preloading efficiency.

5.   **Claims 1 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable**

over Boland et al. (5,872,972) in view of Custer (Inside Windows NT) and further in

view of Admitted Prior Art (Applicant's Specification page 2 line 20 through page 3

line 7).

With respect to claims 1 and 11, Boland et al. teaches a plurality of tasks

(processes) of more than one application (col 4 line 19, "all runnable processes in global

run queue 24, ..."), computing nodes (col 3 line 52), a plurality of local processes (col 4

line 21, "processes which have been previously run and are now affinitized to a specific

processor." Col 4 line 59, "once a processor runs a process, it would never age away its

affinity from that processor."), providing application information to scheduler means

(The scheduler 90 obviously takes information from applications or processes to schedule

the processes into the global priority queue. In column 3 lines 1-5 processes provide

affinity information to the scheduler.), and dynamically creating a schedule of a plurality

of tasks utilizing priorities (col 4 lines 22-24), a local scheduler (Column 4 lines 26-31

teach that each 'processor' consults the global queue. This inherently requires local code

or 'scheduler' to perform the queue analysis. Referring to Figure 7, column 7 lines 26-

28, "These processes may thereafter be reordered based upon process priority within a

nodal priority run queues 71 and 77 ..."), means for ascertaining which process(es) are

assigned to the tasks (Custer teaches on page 84 that a process must be combined with a

thread of execution before it can do any work. The thread of execution "is the entity

within a process that the NT kernel schedules for execution. Without it, the process's

program cannot run." Therefore, it would have been obvious to one of ordinary skill in

the art at the time the invention was made to incorporate means for correlating or

ascertaining which process(es) are assigned to the tasks, because Custer recognized

without the correlation between processes and threads the process or program would not

be able to execute.), and means for prioritizing the processes according to the prioritized

schedule (Applicant has divulged on pages 2-3 that the AIX™ operating system assigns a

common priority to the process(es) required for (or correlated to) a task. Having the

processes, associated with individual tasks, assigned priorities corresponding to the

priorities of the schedule would have been a highly desirable feature in the art.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the

invention was made to incorporate the local scheduling/correlating means of IBM into the

task scheduling system of Cameron et al., because prioritizing local processes according

to the task correlation would have been expected to result in higher cache preloading

efficiency.).

Response to Amendment filed September 1, 1999

Section 112: In light of Applicant's amendment, the *35 USC § 112* rejection set forth in

the previous office action is withdrawn. (Examiner notes that Applicant suggested that

claim 15 depends upon claim 13 when, in fact, claim 13 does not fall in the chain of

dependence.)

Section 103: Applicant has presented the following arguments:

Applicants wrote on page 6 in the amendment mailed September 1, 1999, "The

present invention, in contrast to Cameron, provides for multiple tasks of multiple applications

being active on the multiple processes of any given node at a time." Briefly stated, Applicant has

limitations in amended Independent Claims 1 and 11 of <u>dynamic creation</u> of a task schedule,

ascertaining which tasks are assigned to each node, and prioritizing local processes according to the schedule. Applicant does not appear to provide limitations in Claims 1 and 11 relating to multiple tasks being active or threads of execution executing in the local processes.

Applicants wrote on page 6 in the amendment mailed September 1, 1999, "Applicants contend that the Cameron teachings do not teach or suggest application *programs running on different layers at the same time*; particularly in light of the aforementioned quotations from Column 2, lines 53-58 and Column 7, lines 37-42. ... clearly Applicants are teaching and claiming the assignment of a schedule of priorities and *the local implementation* and scheduling according to that schedule of priorities for tasks of multiple application programs assigned to the same layer." [emphasis added]

Then on pages 7-8 Applicant states, "... it is clear from the description (see: e.g., Page 8, lines 6-7 and Page 12, lines 10-13) that the *different processes on a node can*, as a result of the Invention, *be working on different applications at any given time*; and, not simply that different applications can be loaded on the node at one time while only one application is executing." [emphasis added]

It is noted that the features upon which applicant relies (i.e., multiple tasks being active, programs running on different layers at the same time, the local implementation, and processes working on different applications at any given time) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification,

limitations from the specification are not read into the claims. *In re Van Guens*, 988

F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Conclusion

Any inquiry concerning this communication should be directed to **Gary Fourson**

at telephone number **(703) 305-4392** or E-mail at the address **gary.fourson@uspto.gov**.

Any inquiry of a general nature or relating to the status of this application should

be directed to the Group receptionist whose telephone number is (703) 305-9600.

The fax numbers for formal (703-308-9051), to be intended for entry into the

application, or informal (703-305-9731) communications may be utilized for expedited

transactions.

The Examiner requests that your amendment response including all pending claims

be in paper form accompanied by a 3½ inch IBM format floppy disk which contains a

file copy of your amendment response in WordPerfect, Microsoft Word, or in ASCII text

format. Only the paper copy will be entered. Your 'electronic' file will be considered a

duplicate copy, and signatures are not required on the disk copy. The floppy disk copy is

not mandatory, however, your cooperation is appreciated.

gsf

September 29, 1999

ALVIN E. OBERLEY
SUPERVISORY PATENT EXAMINER
GROUP 2700